REMARKS

Applicant is in receipt of the Office Action mailed February 26, 2004. Claims 1, 4, 5, 7, 12, 14, 16, 18-19, 24, 26, 29, 31, and 37 have been amended. New claims 43-49 have been added to more fully characterize the invention. Thus claims 1, 3-7, and 9-49 are pending in the application. In addition to the remarks presented in a prior Response to the Final Office Action dated May 26th, 2004, which is hereby incorporated by reference, further consideration of the present case is earnestly requested in light of the following remarks.

TELEPHONE INTERVIEW SUMMARY

Applicant thanks Primary Examiner Thai and Examiner Knoll for the courtesies extended during the telephone interview between the Examiners, Mark Williams, and Martin Wojcik on July 15, 2004. During this interview, Examiner Knoll has agreed that by qualifying the trigger to be of asynchronous nature as well as explicating the differences between the system of Rao and the system as claimed in the Application, the Application would come under consideration of being approved. The independent claims have been amended and differences from the related art are expounded below.

Claim 1

Amended independent claim 1 recites:

1. A system for synchronizing a Controller Area Network (CAN) interface with a peripheral device, the system comprising:
a peripheral device, coupled to a host computer via an I/O bus;
a CAN interface, coupled to the host computer via the I/O bus, wherein the CAN interface comprises:
a memory configured to store program code;
an embedded processor coupled to the memory, and configured to execute the program code;
bus interface logic coupled to the embedded processor; and
CAN interface logic coupled to the embedded processor and configured for interfacing with a CAN bus; and
an interconnecting bus, coupling the peripheral device to the CAN interface via the bus interface logic of the CAN interface;

wherein the peripheral device is operable to generate an asynchronous trigger on the interconnecting bus in response to a peripheral event;

wherein the CAN interface is operable to receive the asynchronous trigger via the interconnecting bus;

wherein the embedded processor is operable to execute the program code to perform a CAN event in response to said CAN interface receiving the asynchronous trigger on the interconnecting bus from the peripheral device, wherein the CAN event is performed substantially synchronously with the peripheral event; and

wherein the generation and receipt of the asynchronous trigger, and the performing the CAN event are performed independently of the I/O bus.

Claim 1 recites a system where a peripheral device and a CAN interface are each coupled to a host computer via an I/O bus. The peripheral device and the CAN interface are coupled via an interconnecting bus. The peripheral device is operable to send an asynchronous trigger using the interconnecting bus to the CAN interface upon an event on the peripheral device. The CAN interface, upon receiving the asynchronous trigger via the interconnecting bus, is operable to perform a CAN event that is substantially synchronous with the event performed by the peripheral device. In other words, the asynchronous trigger is operable to synchronize events between the CAN interface and the peripheral device. Furthermore, the generation and receipt of the asynchronous trigger, and the performing the CAN event are performed independently of the I/O bus.

The system of Rao (US 2003/0028701, "Rao") is significantly different from the system of claim 1. Rao describes a real-time performance monitoring facility in an integrated circuit (IC) data processor for monitoring events related to different bus activity. In paragraph [0017], Rao teaches a system on an Integrated Circuit (IC) where:

"A host system 108 is coupled to the first bus 112, while a number of I/O devices 116 and 117 are coupled to the second bus 118. Examples of I/O devices are network adapter cards and disk controllers. The data processor 104 includes first and second bus interfaces (not shown) to each of the first and second buses. The first bus 112 and second bus 118 may be treated by the I/O devices as one logical bus with the help of a bridge 138. An internal bus 122 that may or may not extend outside the data processor 104 is used for communication between a subsystem processor 126 and a local memory 127 via a memory controller 128. The local memory 127 normally stores the instructions to be executed by the subsystem processor 126. Although shown here as integrated on the same die as the data processor 104, the subsystem processor 126 may alternatively be implemented as

a separate die. Communication between the first and second buses and the internal bus 122 is accomplished via a number of bridge-like devices being primary and secondary address translation units P_ATU 130 and S_ATU 134. Data transfers from devices on the first bus 112 and second bus 118 to the local memory 127 may also be achieved using the respective direct memory access (DMA) channels 144 and 146." (Rao [0017]) (emphasis added)

Rao teaches an IC that has a first bus that couples to a host system and a second bus that couples to one or more I/O devices. The IC of Rao also has an internal bus that couples to a memory controller and a subsystem processor. The IC of Rao includes a Data Processor that has monitoring facility for the first and second buses as well as the internal bus:

"The data processor 104 includes an on-chip performance monitoring facility 142. The dotted lines between the block representing the monitoring facility 142 and the respective buses represent signal paths that carry event information from the buses and their respective bus interfaces (not shown) to the monitoring facility 142. The events are occurrences and durations of bus activity which are caused by communication between devices on the internal bus 122 (e.g., subsystem processor 126 or memory controller 128), devices on the first bus 112 (e.g., the host system 108) and devices on the second bus 118 (e.g., the I/O devices 116 and 117). Some useful events to be monitored include bus idle and data cycles, number of grants, number of retries, device acquisition time, and device ownership time." (Rao [0018]) (emphasis added)

The monitoring facility of Rao is operable to use counter registers to record a number of times a certain event occurred. That counter register may have a predetermined maximum value that in turn may cause an interrupt that is interpreted by software executing on the subsystem processor 126 of Figure 1 of Rao:

"Another embodiment of the monitoring facility 142 as shown in FIG. 4 is one having a counter status register 414 that allows software to identify an event counter 166.sub.i that may have overflowed. The status register is software-readable and is used to identify those event counters 166.sub.i that have reached predetermined counts, and in particular an overflow condition. The software may use this information to manage a performance monitoring session as described below in the embodiment of FIG. 5. The counter status register 414 in one embodiment includes a number of one-bit latches each receiving a notify signal from a respective event counter 166.sub.i. Each counter 166.sub.i can generate a notify signal in response to reaching a predetermined count while counting events. For instance, a predetermined count may be the maximum count value of the event counter 166.sub.i, where the notify signal in effect becomes an overflow indication that the counter has reached its maximum count value. The notify

signals are fed to OR logic 418 which has a number of inputs 1 . . . N each coupled to a respective one of the latches that are part of status register 414. The OR logic 418 forwards the notify signal to be interpreted by the software. This may be done by routing the notify signal to an interrupt input of the subsystem processor 126, such that when the signal is asserted, software causes the processor 126 to read the contents of the counter status register 414 to determine which one of the event counters 166.sub.i has reached its predetermined count. As explained below in connection with FIG. 5, such a feature is useful for monitoring a large number of events using fewer event counters 166.sub.i by reading and storing intermediate values of the event counters 166.sub.i" (Rao [0028]) (emphasis added)

However, in the system of Rao, it is the system processor that receives an interrupt from the monitoring facility. In all embodiments listed by Rao, the other devices read the one or more registers 170 of the monitoring facility 172 of Figure 1:

"The monitoring facility 142 includes a number of event counters 166.sub.1, 166.sub.2, . . . , 166.sub.N (166.sub.i) and a corresponding number of counter registers 170.sub.1, 170.sub.2, . . . 170.sub.N (170.sub.i). Each of the registers 170.sub.i contains the current count value of its corresponding event counter 166.sub.i. The registers 170.sub.i may be read by software being executed in the system 100. The software in one embodiment may comprise instructions stored in the local memory 127 for execution by the processor 126. The registers 170.sub.i may be a series of memory-mapped registers that are accessed by the software via the internal bus 122. Other configurations for accessing the registers 170.sub.i, however, are possible. For instance, the registers 170.sub.i may be accessed via the first bus 112 and the P_ATU 130, in response to instructions stored and executed by the host system 108." (Rao [0019]) (emphasis added)

Rao teaches a system where the software executing in the system 100, upon being interrupted by the monitoring facility, is operable to perform an action. Alternatively, Rao teaches a system where the host system 108 is operable to access registers 170 using the first bus 112.

In contrast, claim 1 recites that an event on a peripheral device is operable to asynchronously trigger an event on a CAN interface using an interconnecting bus. Rao does not teach or suggest analogous relationships between devices on the IC of Figure 1. For example, Rao does not teach that an event on a first device is operable to asynchronously trigger an event on another device, such as the I/O device 116 on the second bus or the memory controller 128. Instead, Rao teaches that the monitoring facility 142 is operable to count events occurring on the first bus, and upon reaching a

predetermined count, either assert an interrupt to be read by the subsystem processor 126 or polled by the host system 108. In the system of Rao, the monitoring facility 142 counts events on the first bus or on the second bus. Neither the host system 108 nor the memory controller 128 nor the subsystem processor 126 nor the first I/O device 116 nor the second I/O device 118 is able to trigger an event on any other device directly. Instead, as mentioned above, the monitoring facility 142 on the IC is operable to count events using event counters 166 and record the number of events in a corresponding register 170 of the monitoring facility. Applicant further notes that the device/bus topology expressed in claim 1 is not found in Rao.

In actuality, the monitoring facility of Rao teaches away from the peripheral device of claim 1 asynchronously triggering the CAN interface using an interconnecting bus. The peripheral device of claim 1 asynchronously triggering the CAN interface using an interconnecting bus does not use an external entity, i.e., the monitoring facility, to count events. Claim 1 does not teach counting events, but instead teaches directly and asynchronously triggering an event on the CAN interface by the peripheral device. The asynchronous trigger signal is sent over the interconnecting bus to the CAN interface from the peripheral device, independent of the I/O bus. Rao does not teach or suggest an analogous system or mechanism.

Furthermore, Applicant respectfully submits that although the monitoring facility 142 of Rao is operable to raise an interrupt upon one of the counter registers reaching a predetermined count value, this is different from generating an asynchronous trigger by the peripheral device. The peripheral device of claim 1 is operable to generate the asynchronous trigger and send the trigger to the CAN interface using the interconnecting bus. As noted above, claim 1 does not use another entity, i.e., the monitoring facility 142 of Rao, to monitor, count, record in a corresponding counter register, and either raise an interrupt to a separate subsystem processor 126 or be polled by the host system 108.

Specifically, in paragraphs [036] and [037] Rao describes the architecture and use of the event counters. Rao describes a circuit where a counter updates its own counter value upon receiving a 'qualified counter value' signal from an 'event qualification

logic'. The event qualification logic operates to choose the proper clock domain using domain qualifier circuits, which operate to indicate the corresponding clock domain. The value in the counter register of Rao is operable to be used by the monitoring software, as described in paragraph [015]. Rao describes a system where the monitoring software can either poll or be interrupted by the counter reaching a pre-determined count. The monitoring software is operable to perform an operation based on the type of the interrupt. Thus Applicant notes that the event counters of Rao teach away from the peripheral device of claim 1 asynchronously triggering the CAN interface using an interconnecting bus.

Therefore, for at least the reasons presented above, Applicant respectfully submits that claim 1 is patentably distinct over the cited art.

Similar arguments apply with equal force to independent claims 7, 14, 19, 24, 31, and 37. Applicant thus respectfully submits that each of the independent claims 1, 7, 14, 19, 24, 31, and 37, and claims dependent thereon, are patentably distinct over the cited art, and are thus allowable.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

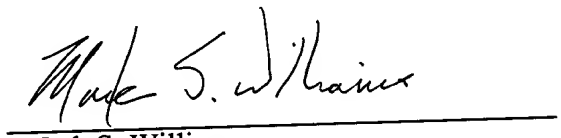
In light of the foregoing amendments and remarks, Applicant submits the application is now in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-22400/JCH.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

Respectfully submitted,



Mark S. Williams
Reg. No. 50,658
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: 7-22-04 JCH/MSW /MRW